



TITLE OF THE INVENTION

SYSTEM FOR PERFORMING COUPLED FINITE ANALYSIS

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims priority to United States Provisional Patent Application Serial No. 60/246,399, filed November 7, 2000, now abandoned.

BACKGROUND OF THE INVENTION

Field of the Invention

[0002] The invention pertains to the field of software, computer systems and related methods including models that operate through the use of finite analysis, finite volume, and/or finite difference techniques.

Description of the Related Art

[0003] Finite element, finite volume, and finite difference modeling techniques can generally be described as mathematical approximations of often-complex problems that represent physical behavior. The mathematical models are useful in designing physical apparatus or systems and in predicting the behavior of existing apparatus or systems. These models use a mesh or grid that is superimposed over the system being studied to provide a plurality of cells or elements. These elements may be modeled in multiple dimensions, for example, one, two, or three dimensions. Mathematical equations that represent or approximate physical or quantitative behavior are applied to each cell with the resultant formation of a system of equations that are expressed as matrices, and that are solved using generally known techniques of linear algebra. Such mathematical techniques commonly involve iterating through a set of equations until a threshold convergence is achieved, i.e., until the difference between successive iterations through a system of mathematical approximations becomes so small that it is suitably negligible with respect to the exact or rigorous solution of the system of equations being modeled. The term "finite analysis" is hereby defined to include finite element, finite volume and finite difference models.

[0004] A variety of patents have issued on various finite element and finite difference techniques. For example, U.S. Patent Nos. 5,956,500 and 5,901,072 pertain to a method for incorporating boundary conditions into a finite analysis model. These patents disclose generating a

finite analysis model having a finite element shim interposed between a test bar and a ground, where the characteristics of the shim are selected based upon measured natural frequencies of the test bar. U.S. Patent No. 5,768,156 addresses a method of automatic mesh generation for finite analysis. The meshes are generated using whisker chords to form all-hexahedral elements. Similarly, U.S. Patent No. 5,731,817 pertains to a system that generates a hexahedron mesh and then performs finite analysis on the mesh. U.S. Patent No. 5,581,489 discloses a model generator including data input for an object to be modeled, a material information generator providing material properties for the object being modeled, a mesh processor for generating a mesh, and an output generator coupled with a finite analysis processor. U.S. Patent Nos. 5,553,206 and 5,315,537 pertain to automatic mesh generation systems.

[0005] Finite analysis programs that provide solutions to specific problems are commercially available. For example, ABAQUS is available from Hibbitt, Karlsson and Sorenson of Pawtucket, Rhode Island to model structural mechanics and nonlinear heat transfer. ANSYS is available from Ansys, Inc, of Canonsburg, PA to model structural mechanics and heat transfer. ASTMA is public domain software available from the National Aeronautics and Space Administration (NASA) that models heat transfer and ablation. IDEAS is available from Structural Dynamics Research Corporation of Milford, Ohio to provide pre and post-processing images of the model. SINDA from SINDA, Inc. of Tempe, Arizona models heat transfer. TEX CHEM models chemical reactions and chemical equilibrium. RECESS is a program developed by Thiokol Propulsion of Brigham City, Utah to model internal ballistics. CDCA is a computational fluid dynamics program developed by Pennsylvania State University to model crack combustion where a fracture in a propellant affects burn condition. CCM is a similar computational fluid dynamics program available in the public domain, and is available from the Air Force Research Laboratory (ARFL).

[0006] Many specific examples of the need for finite analysis programs exist, for example, in the field of rocketry and missile maintenance. In fact, the commercial finite analysis programs that are mentioned above have many specific applications in this field. For example, the public domain ASTMA program and derivatives thereof can be used to model the burning away of material from a rocket engine nozzle.

[0007] A problem exists in the field of finite analysis modeling because engineering specialties do not encompass a wide array of specialized problems that are presented by complex physical situations. For example, the burning of a solid fuel rocket motor presents a multifaceted problem including structural mechanics, material properties, internal ballistics, chemical reactions, heat transfer, crack combustion, and fracture mechanics. An engineer who is modeling only one of these problems using a commercially available or proprietary finite analysis program for this purpose may require a full year just to become proficient at using the package. Such engineer is typically not trained in more than one or two of the specialty problem areas and is often incapable of running models in areas outside his or her area of expertise. Very few, if any, engineers succeed in acquiring the training that is required to model all aspects of this problem, and a team of modelers often is required to produce modeling results through a laborious process involving the transfer of model results between different engineers and/or finite analysis codes.

[0008] It is typical in the finite analysis art that there exist separate programs to model computational fluid dynamics, structural mechanics, heat transfer, internal ballistics, etc. This segregation of problems exists, in part, due to the lack of overlap in specialty areas as described above, but it also exists because the situations encountered for actual modeling purposes are very diverse and require flexibility if the model is to have optimum results. A great deal of effort may be expended to develop a comprehensive model where the usefulness of the model diminishes with its complexity.

[0009] The foregoing problem is normally addressed by a sharing of data between engineers or engineering groups that encompass multiple specialties. This sharing of data leads to additional problems. An engineer receiving model results from another engineer for further processing does not necessarily understand the model results that he or she has received, and this circumstance can lead to computational error. For example, the preceding engineer may provide results from a less thorough model than is required for optimum results in subsequent calculations, or problems may arise through the nodes of meshes being at different locations when data is passed from a first model to a second model.

[0010] Special problems also arise when an engineer receives prior calculation results and uses them as input in a subsequent model addressing a different problem because subsequent calculation results may affect input for the prior model. For example, an internal ballistics program

may be used to calculate internal pressures in a solid fuel rocket motor. These pressures are subsequently used in structural mechanics calculations where the rocket fuel deforms in a visco-elastic manner. The volume changes from the structural mechanics solution have significant effects upon the internal ballistics results which, in turn, affect the structural mechanics model. Thus, a repetitive sharing and transfer of computational results is required from successive iterations until the effects of the separate programs upon one another between different runs become negligible. Furthermore, the respective modelers may even be unaware that their individual model or an aspect of their model results can affect other models that provide results including input data for subsequent models.

BRIEF SUMMARY OF THE INVENTION

[0011] Accordingly, an object of the present invention is to provide a finite analysis modeling system that permits the user to identify a joint problem for coupled solution through the use of a graphical user interface or a scripting language.

[0012] Another object of the invention is to provide a single interface that ties or couples together a plurality of finite analysis programs for purposes of solving complex problems while maintaining flexibility.

[0013] Another object of the invention is to provide data linkages for coupling programs in a predetermined manner to facilitate coupling for the user.

[0014] Yet another object of the invention is to facilitate user-specified complex solutions without requiring the user to understand all aspects of each discipline.

[0015] These objects and advantages, as well as other objects and advantages will be apparent to those skilled in the art upon reading the attached drawings together with the accompanying specification.

[0016] A computer system, as well as associated software and methodology are provided for obtaining a joint solution through the use of multiple finite analysis programs. The system, software and methodology tie or couple together a plurality of finite analysis programs for purposes of solving complex problems while maintaining flexibility through the use of a graphical user interface.

[0017] Preferred embodiments operate in an environment of use where there is memory storage for first and second finite analysis programs. The first finite analysis program acts upon first program input values to provide first program output values based upon the first program input values. The second finite analysis program acts upon second program input values including a first joint data set having at least a subset of the first program output values. The second program output values include a second joint data set of values that can be used as first program input values. One of the first and second finite analysis programs is a computational fluid dynamics program.

[0018] As mentioned above, the software and the method, as well as the computer system in an electronically programmed state, all utilize a graphical user interface that is operable to identify a joint problem that both the first and second finite analysis programs can jointly and in combination solve, and to specify at least one criterion for a joint solution. Data processing linkages between the first and second programs, and the user, provide the first program input values to the first program. Embedded commands in the graphical user interface or related programs execute the first finite analysis program to obtain the first program output values including the first joint data set. Data processing linkages between the first and second programs then provide the second finite analysis program with second program input values including the first joint data set. Embedded commands in the graphical user interface, related programs, or a scripting language execute the second finite analysis program to provide second program output values including the second joint data set. Data processing linkages between the first and second programs then provide the first finite analysis program with first program input values including the second joint data set.

[0019] An aspect of the software, system, and method is that the criterion for the joint solution specified through the graphical user interface preferably includes an iterative convergence criterion for threshold convergence of the joint solution. The program instructions, as well as the methodology of operation, then include repeating the steps of providing data and executing the first and second programs until the specified threshold convergence criteria is achieved. This iteration, which is performed outside the boundaries of the first and second finite analytical programs, is performed automatically and without user intervention after the initial data input and execution of the first and second programs.

[0020] Another aspect of the software, system and method is that the graphical user interface is optionally used to identify the joint problem by selecting the joint problem, in

combination with the computational fluid dynamics problem, to preferably include an additional finite analysis problem selected from the group consisting of structural analysis problems, heat transfer problems, chemical reaction problems, chemical equilibrium problems, internal ballistics problems, and fracture mechanics problems.

[0021] A third, fourth, fifth or additional programs may be selected in like manner and placed in the coupled or joint program execution for convergence among all of the programs. For example, where a third program is selected for use in the coupled solution, the third program acts upon third program input values selected from the group consisting of first program output values, second program output values, and combinations thereof to provide third program output values. The third program output values include a third joint data set comprising input values selected from the group consisting of first program input values, second program input values and combinations thereof. The method of operation then includes executing the third program to produce the third program output values including the third joint data set. The third joint data set is provided, as needed, providing the third joint data set to the first and second finite analysis programs with corresponding input values selected from the group consisting of first program input values, second program input values and combinations thereof. Iterative threshold convergence can then be achieved according to convergence criteria specified through the graphical user interface.

[0022] The software, system and method can be applied to a number of problems, for example, in the field of missile design and maintenance. For example, where a solid fuel rocket has maintenance operations performed on it, and these operations provide computed tomography results showing a crack in the propellant, the effect of this crack may be modeled to determine whether the crack will prevent the missile from completing its intended purpose if launched. In this case, the computational fluid dynamics program may be a crack combustion program, the system provides means for modeling crack combustion in a missile based upon computed tomography taken from a missile, and the other of the first and second programs is, by way of example, a structural analysis program.

[0023] The finite analysis may be supplemented through use of a programming language, especially an extensible object oriented scripting language that is capable of issuing calls to a variety of other programming languages. In this manner, a software screen or shell may be built to provide, for example, a function library that duplicates the GUI functionality using direct commands in the

guise of scripted functions which operate across the boundaries of multiple operating systems, multiple programming languages, and/or data translations, as required to couple a plurality of programs. The scripting language preferably has looping and decisional logic capabilities giving the user infinite control over the operation of a variety of coupled finite analysis programs. The scripting language and its associated function library may optionally be accessible from the GUI.

BRIEF DESCRIPTION OF THE DRAWINGS

[0024] Figure 1 is a schematic diagram of a preferred system embodiment according to one aspect of the invention;

[0025] Fig. 2 is a process schematic operational diagram for the system of Fig. 1;

[0026] Fig. 3 depicts a mid-sectional view of a rocket motor having various defects that are modeled by finite analysis causing mesh boundaries to change, in order to provide additional detail with respect aspects of the preferred method illustrated with respect to Fig. 2;

[0027] Fig. 4 is a schematic representation of menu options for a preferred graphical user interface that operates according to another aspect of the invention;

[0028] Fig. 5 depicts a selected model that can be created using a scripting language to couple a plurality of finite analysis programs;

[0029] Figs. 6—11 are part of the interactive online user's manual of Appendix 1; and

[0030] Figs. 12—92 comprise the computer program listing of Appendix III.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS AND METHODS

[0031] According to one of the various embodiments of the invention, there is now shown a schematic computer system that is programmed to an electronically configured state through the use of software for the performance of a method, as described above. The system, software and methodology tie or couple together a plurality of finite analysis programs for purposes of solving complex problems while maintaining flexibility through the use of a graphical user interface. By "coupling" it is meant that the results from one finite analytical model are provided to another model as input. For example, the results from a first model may be provided as input to a second model, and the results from the second model may optionally be provided as input to the first model. This process may be repeated until a user-specified convergence is achieved.

[0032] Fig. 1 depicts a schematic diagram of a computer system 100 including a central processing unit (CPU) 102 that is linked to a magnetic or optical storage medium 104, a display 106, such as a flat panel plasma display or cathode ray tube, and input/output devices 108, such as a computer keyboard, a pointing device such as a trackball or mouse, other computers, and additional magnetic or optical storage media. The hardware components 102 through 108 of system 100 may comprise those that are found in most personal computers and engineering workstations.

[0033] As shown in Fig. 1, software from storage media 104 or I/O devices 108 has been used to program a graphical user interface (GUI), which is represented by a GUI RAM block 110. The GUI 110 is used to interface with a user through I/O devices 108 to identify a plurality of programs for coupled execution. These programs are represented by RAM blocks 112-120. Block 112 includes a first finite analysis program that provides a computational fluid dynamics solution. Block 114 includes a second finite analysis program that provides a structural solution. Block 116 includes a third finite analysis program that provides a heat transfer solution. Block 118 includes a fourth finite analysis program that provides a ballistics solution. Block 120 includes a fifth finite analysis program that provides a mesh generator program. The mesh generator program is not used in iterative convergence, but it is used to generate individual meshes for use in each of the first, second, third, fourth and fifth finite analytical programs.

[0034] The system, software and method preferably operate in an environment of use where there is memory storage for the first and second finite analysis programs corresponding to blocks 112 and 114, for example. The first finite analysis program acts upon first program input values to provide first program output values based upon the first program input values. The second finite analysis program acts upon second program input values including a first joint data set having at least a subset of the first program output values. The second program output values include a second joint data set of values that can be used as first program input values. One of the first and second finite analysis programs is a computational fluid dynamics program.

[0035] An aspect of the software, system, and method is that the criterion for the joint solution specified through the graphical user interface preferably includes an iterative convergence criterion for threshold convergence of the joint solution. The program instructions, as well as the methodology of operation, can then include repeating the steps of providing data and executing the

first and second programs until the specified threshold convergence criteria is achieved. This iteration, which is performed outside the boundaries of the first and second finite analytical programs, is performed automatically and without user intervention after the initial data input and execution of the first and second programs.

[0036] Fig. 2 is a block diagram of a process 200 representing the operation of system 100. Step 202 entails using the GUI 110 to identify a problem for coupled solution between two or more of the programs shown in RAM blocks 112-118. Data linkages, as well as a sequence of operation for the respective programs to solve a coupled solution, have been previously entered by an expert or team of experts in coupling the finite analytical programs.

[0037] Data for these solutions is provided to the system 100 in step 204 where, for example, computed tomography data from missile maintenance operations may be provided as input for a structural model. Additional data including such data as materials properties; the specification of materials; boundary conditions of temperature, pressure, force, and any other useful data, is provided as needed by the specific analytical programs. A mesh generator program, such as the fifth program 120 or a plurality of such programs designed for specific applications, is also used to provide data input in the form of mesh generation.

[0038] Step 206 preferably begins once the data input of step 204 is concluded, as shown in Fig. 2, but the process 200 may also interrupt itself to ask the user for input at any time. The first program is executed in a sequence of execution designed by the expert or team of experts. The execution of the first program in step 206 provides first program output including a joint data set that may be shared, in step 208, as input data with the second finite analysis program shown in block 114. Additional data subsets may be generated and shared with any other of the finite analysis programs in blocks 116 and 118, for example. The second finite analysis program is executed in step 210 with a similar sharing of data in step 212. In step 212, however, the second program output may provide a second joint subset of data that can be used as input data for the first program once step 206 is executed again. The remaining programs are executed in a similar manner with appropriate data linkages being provided in cooperation with executable code associated with the GUI 110 so that the user does not need to specify data linkages to obtain a coupled solution.

[0039] In preferred embodiments, a portion of the data input in step 204 includes a criterion or criteria for iterative threshold convergence. For example, where the ballistics block 118

produces pressure data that modifies the program input for the structural block 114 due to the elastic deformation of rocket propellant, the initial boundary condition of pressure in the structural block 114 may be modified with time, as may the pressure conditions of the computational fluid dynamics model 112. The change in pressure from the ballistics model causes the structural volume results to change, as computed by the structural block 114. In turn, the computational results from the structural block 114 including an altered volume may be supplied as input to the computational fluid dynamics block 112 to obtain still different pressures. Both the volume results from structural block 114 and the pressure results from the computational fluid dynamics block 112 may, in turn, be supplied as input to the ballistics block 118.

[0040] In step 216, this iterative procedure continues with repetitive iterations through steps 206, 208, 210, 212, and 214 until the specified convergence criteria or a maximum number of iterations representing a probable divergent solution is achieved. For example, the user-specified convergence criteria may indicate convergence when the total pressure change for computations in the ballistics block is less than a fixed value, e.g., one-half psi between successive iterations; when the volume change in successive iterations through the structural block 114 is less than a predetermined delimiting percentage of the total volume; when calculations for the same value obtained as output from different programs match within a delimiting percentage; combinations of these examples; and any other useful convergence criteria.

[0041] Where convergence is determined to have been met in step 216, a post-processing step 218 stores the results, provides a visual display of the results on the display 106, and prints the results as needed. Step 218 also entails any other desired post-processing step.

[0042] Step 220 entails the interpolation or projection of meshes for reasons that are illustrated, by way of example, in Fig. 3. In summary, mesh boundaries frequently move as a result of physical responses to system stimuli, e.g., heat, pressure, strain, and ablation, so that the model boundaries must be adjusted due to these movements. Fig. 3 depicts a section 300 of an aging cylindrical solid fuel rocket motor. The motor includes a composite outer shell 302, a liner 304 that is used to protect the outer shell from burn-through during launch, an inner core 306 of visco-elastic rocket propellant, and an interior core 308 comprising a burn chamber. Conventional maintenance operations including computed tomography have diagnosed an area 310 of debonding between the inner core 306 and the liner 304. Computed tomography has also diagnosed a crack 312 that is

growing in the aging rocket propellant. Finite modeling has provided a coupled solution involving internal ballistics, structural, and computational fluid dynamics to demonstrate various flow regimes including regimes 314, 316, 318, 320, and 322. A few rocket motors have been known to explode due to cracking of the propellant as shown in crack 312.

[0043] The propellant region 324 tends to deform more readily due to higher velocity downstream of crack 312, which results in a higher upstream pressure. Flow conditions around the crack 312 have a Bernouli effect that results in decreased pressure in flow regime 320 downstream of the crack 312. The regime 320 narrows the flow through regimes 316 and 318, and imparts increased velocity. Flow regime 322 is a relatively low pressure flow regime. Another potential problem is that the inner core 306 may strip away from the debond area 310 with disastrous results.

[0044] The computational fluid dynamics model begins calculation using a cylindrical mesh (not shown in Fig. 3) having an outer radius equal to the inside diameter of the inner core 306. This cutter radius is shown as lines 334 and 336 in Fig. 3. As shown in Fig. 3, the computational results from a first pass iteration of the coupled programs, which moved the inner diameter of the inner core 306 out to lines 326 and 328, primarily due to deformation of the inner core 306. Convergence has not yet been achieved, so it becomes necessary to adjust the associated meshes and relevant boundary conditions to account for the deformation of inner core 306. Boundary condition projection is performed by projecting the results from the fluid mesh at lines 326 and 328 to new boundaries at lines 330 and 332. Pressure results for CFD, for example, may be projected to new boundaries of the structural mesh at lines 330 and 332.

[0045] Projecting of boundary conditions is done by locating associated positions on the complementary mesh and extracting the necessary data to create comparable boundary conditions on the current mesh, i.e., the mesh that is receiving the boundary conditions. Usually, the associated position is perpendicular to the surface of the receiving mesh, but may be the closest point if the perpendicular is not usable. The boundary conditions are then created on the current mesh from data on the complimentary mesh.

[0046] Alternatively, where the initial pressure assumption is too high, the initial mesh for the computational fluid dynamics model may be found having a radius at lines 330 and 332. In this case, the computational results show the propellant having moved radially inboard to lines 326

and 328. In this alternative case, an identical method of projecting boundary conditions can be used.

[0047] Differences between iterative steps may require the boundary of the mesh to be modified. Large differences may require the finite element mesh to be remeshed for use in subsequent calculations. Results and boundary conditions must now be transferred to the modified mesh by interpolation for the analysis to continue. If mesh interpolation is not desired, or if the fracture geometry prevents interpolation, the user may be prompted to assist in the generation of a new mesh over the region of concern, or an entirely new mesh may be generated over the new boundary by a mesh generation package. Techniques for automated mesh generation to discretize a model of a body are generally known. Examples of such techniques are included in the foregoing discussion of related art, e.g., U.S. Patent No. 5,729,670, which is incorporated by reference to the same extent as though fully disclosed herein.

[0048] As mentioned above, the software and the method, as well as the computer system in an electronically programmed state, all utilize a graphical user interface that is operable to identify a joint problem that both the first and second finite analysis programs can jointly and in combination solve, and to specify at least one criterion for a joint solution. Data processing linkages between the first and second programs, and the user, can provide the first program input values to the first program. Embedded commands in the graphical user interface or related programs execute the first finite analysis program to obtain the first program output values, including the first joint data set. Data processing linkages between the first and second programs then can provide the second finite analysis program with second program input values including the first joint data set. Embedded commands in the graphical user interface, related programs, or a scripting language can be used to execute the second finite analysis program to provide second program output values including the second joint data set. Data processing linkages between the first and second programs then provide the first finite analysis program with first program input values, including the second joint data set.

[0049] Fig. 4 depicts a plurality of graphical fields such as may appear, for example, on the CRT of a user who is interacting with the graphical user interface of the preferred embodiment. The graphical elements are also known as fields, and may be accessed, for example, by clicking the button of a mouse to reveal additional menu options or icons that are associated with each field.

[0050] A GUI 400 includes a file field 402, which includes submenu options (in bold font) including functionality to open a new file; open an existing file; close an active file; read data for input to a list-selected finite analysis program and configure that program for options and units; write data for input to a list-selected finite analysis program and configure that data for options and units; translate data from one finite analysis program to another between a list of *read from* programs and a list of *write to* programs; save the current file and save as a new filename; print the data representation on the screen, print preview, print setup, copy the data representation on the screen, and exit the program. In summary, these submenu options permit the user to manage and access the program data files.

[0051] A create field 404 permits the user to interact with a mesh editor to identify the location and type of mesh. The mesh is drawn interactively based upon user-specified parameters up to and including automatic mesh generation. The create field 404 includes submenu options including whether the mesh system is based upon rectangular, cylindrical or spherical coordinates, the reference plane that is viewed on the screen, and the starting coordinates of the mesh. A point option permits the user to enter points based upon numerical coordinates or to enter points by using the mouse to select points. A curve may be drawn as a line, arc, circle or spline. A surface option permits the user to create surfaces from a list of predetermined boundary curves, e.g., spheres, cylinders, and planes, or by extruding or rotating curves. A volume option permits the user to create volumes from a list of predetermined boundary surfaces, e.g., spheres, cylinders, and planes, or by extruding or rotating a surface. A node may be created by entering the location of the desired node. The nodes may be connected by specifying elements that connect the nodes, as well as element geometrical behavior and material type. Nodes may be assigned positional constraints that may also be related to adjacent nodes and free faces. The nodes also have labels and are assigned special graphical representation, e.g., colors. Once a surface or volume has been generated, a mesh option permits the automatic generation of a mesh over the surface of the body or a mesh representing the volume of a body, together with entry of material properties for the mesh or elements of the mesh. Nodes, faces, elements, points, curves, surfaces, and volumes may be treated as a group that shares special properties. Boundary conditions may be defined including pressure, restraint, force, moment, temperature, convection, radiation, heat flux, point source, volume source, specie concentration, specie convection, and specie flux. New materials may be added and material

properties may be entered to a database that tracks material properties. Similarly, chemical species may be added and chemical reactions may be entered into an interpretive equation editor.

Additional tools for displaying the meshes include a plurality of CAD related menu options, e.g., to view a mesh or access geometry creation functions. In summary, the create field provides an interactive mechanism for creating meshes, creating boundary conditions, and assigning chemical and material properties to components of the meshes.

[0052] An edit field 406 permits the user to modify the work plane that is represented on the CRT for viewing. The orientation of the work plane may be designated by coordinate entry, and the character of the work plane can be designated as rectangular or cylindrical. The color and appearance of mesh or body components may be changed, e.g., as by changing the color of points, curves, surfaces, volumes, nodes, and elements. Chemical reaction data may be edited. In summary, the edit field permits the user to change the appearance of an existing mesh without deleting elements of the mesh or body.

[0053] A delete field 408 has the same list of submenu options as does the create field, except the submenu options are used to delete elements, as opposed to creating them. Additionally, boundary conditions may be deleted in sets of related boundary conditions. A list field 410 is used to provide a list of items encompassing the same menu options that exist for the delete field.

[0054] A post-processing field 412 contains submenu options including superposition of calculation results on the mesh; interpolation of results from one mesh to another as specified by model names for the meshes; interpolation to nodes, or centroids, and to average results; deformed geometry may be shown to compare the degree of deformation in a system component as brought about by the predictive model results; contour plot, vector plots, and X-Y plots of predetermined data may also be obtained.

[0055] A model field 414 permits the user to identify a selected model for the coupled solution of a joint problem. A tools field 416 includes a coincident node check that identifies nodes having locations within a predetermined or user-specified tolerance of one another. An element distortion check option permits the user to specify which control measures for distortion the system will consider and to input the desired control values including control measures for Gauss point distortion, aspect ratio, area ratio, quad angle ($90^\circ +/-$), triangle angle ($45^\circ +/-$), and element warping.

[0056] A view field 418 has submenu options including selected adjustments to display settings including color representation of diagram components on the display; blanking and unblanking of points, curves, surfaces, nodes, volumes and elements on the display; dynamic view including pan, magnify, rotate, 3D rotate, and zoom options; reset of the dynamic model changes including a return to the originally depicted conditions; autoscale in which the scale of the model representation on the display is adjusted to reveal the entire model in optimum context; view orientation in which the display on the screen is changed to represent an eye orientation from a user-entered set of coordinates, rotation of the screen perspective a specified distance about a user-selected coordinate, and continuing rotation of the model representation about a user-selected axis.

[0057] The help field 420 provides hyperlink access to user information, such as the information that is shown in the attached appendix.

[0058] Fig. 5 depicts a preferred process 500 for solving a joint problem of a type that may be selected from a menu option using the GUI 400, namely, a structural analysis ballistics problem. Geometry data involving the interior of a missile is obtained using conventional computed tomography techniques for nondestructive evaluation (NDE) of the motor core. This data is provided in step 502 as a body for modeling situations of the type shown in Fig. 3, for example. Step 504 entails automated meshing and may be performed using any number of conventional automatic meshing programs. In step 506, a processor receives user-selected input data and executes program instructions for the first analysis program, which in this case happens to be a structural analysis program modeling time dependent nonlinear visco-elastic (NLVE) deformation of the rocket motor as the interior propellant burns. Structural integrity is checked in step 508 to assure that the motor has not exceeded design specifications or a failure point.

[0059] The NLVE output includes deformations, stresses and strains, which are provided as second program input values to a second program, which comprises a fracture analysis or fracture propagation program that is used in step 510. The output from the fracture analysis program includes deformation and fracture propagation with automated meshing of the propagation-deformed system. A burnback analysis is executed in step 512 including an approximation of the burning surface area that is modified or deformed through burning in the

rocket motor core. The burnback analysis output includes a time-dependent surface area, and a new mesh may be generated over this area.

[0060] The new surface area and mesh is supplied to a computational fluid dynamics and/or ballistics program as program input data in step 514. Iteration continues along loop 516 until convergence is achieved and performance data is supplied in step 518. The computational fluid dynamics program that is used in step 514 may be configured to provide a three dimensional transient solution that has not been previously done in the art. For example, a special scripting language may be used to provide threshold convergence of the solution at small time steps with sequential performance data being provided in step 516 for each time step. Computations of this type may require several CPU weeks to complete, even where the processor is operating in the gigahertz range of clock speeds, and the manual intervention that would have been required to couple the programs for this type of solution was simply impossible using prior methods.

[0061] Other coupled CFD solutions include steady state flow for 1D, 2D and 3D systems in combination with elastic structural deformation or nonlinear visco-elastic structural deformation; quasi-steady state flow in combination with crack propagation; transient 1D, 2D and 3D flow in combination with elastic structural deformation or nonlinear visco-elastic structural deformation; and transient flow with crack propagation.

[0062] It should be noted that any one of the programs which is executed in loop 516 can be a first or second program of the type described herein. The order of execution of the first and second program can be any order because the first and second designation serves merely to distinguish one program from another and does not relate to any specific order of program execution unless otherwise specified. Thus, the computational fluid dynamics program may be executed fourth in order, and this order of execution is still consistent with describing the program as a first or second program. Additionally, even though the respective programs of steps 506-514 are different programs and are not simultaneously executed by a processor, although they may be simultaneously executed in a distributed processing or multitasking environment, these programs are said to *jointly and in combination* provide a solution to the joint problem of structural ballistics analysis because they cooperate in an iterative scheme for a joint solution and/or the programs share data with one another even if no repeat iteration is required.

[0063] Prior mention has been made of a scripting language. The scripting language permits advanced users or system experts to issue program commands that are comparable in analogy to function calls from an object-oriented programming language. A particularly preferred scripting language is Python, which is a copyrighted but freely usable and distributable product, even for commercial use, and is available from PythonLabs at www.python.org.

[0064] The Python scripting language is often compared to other object-oriented programming languages including Tcl, Perl, Scheme or Java, which may provide object oriented substitutes for Python. According to a presently contemplated but merely illustrative embodiment of the invention, the executable code supporting the GUI 400 is written in Python scripting language by an expert in one or more of the finite element analytical programs. An *expert* is hereby defined as a person who has at least five hundred hours of training and program use in a particular finite analysis program, and this time of use is preferably more than one thousand hours.

[0065] An additional GUI element preferably makes available the scripting language to ordinary users who may alter copies of expert-preprogrammed code or write their own code. On networked systems, these user-defined scripting packages may be shared among all users, subject to an expert review and approval process. A Python module was created as an alternative scripted interface comparable to GUI 400, to support geometry creation, finite element models for solving joint problems, data file interfaces and data linkages between different finite analytical programs, and post processing activities. An example of Python code for a data file translation script between two finite analysis programs is:

```
import FEM  
model=doc.NewModel('PlaneEl')  
model.RdldeasMS('RWIdea1.unv')  
model.WrAbaqus('../Work/PlaneEl.inp')
```

[0066] A preprogrammed library of basic functions that duplicate the GUI functionality is provided for user-specified execution in the Python scripting language, which may issue calls to C++ language functions and other languages. The function library may be accessed directly to provide the user with the ability to perform automated calculations without having to interact with the GUI, however, the library is preferably accessed through use of a submenuing or execute option permitting a user to access and/or reprogram the library functions. Classes of functions include

functions that create new documents, open existing documents, save documents, create curves, surfaces, volumes, create boundary conditions, interpolate, read and write files, translate data between applications, as well as any other activity that has been previously described as being practicable through the use of GUI 400. In part because the Python code is extensible to other languages, Python provides tremendous power and versatility including looping and decisional logic capabilities.

[0067] Applicant hereby incorporates-by-reference Appendix I (printouts from an interactive online user's manual for the FEM Builder), Appendix II (a compact disk in duplicate comprising a computer program listing appendix containing computer software according to a presently preferred system and method according to the invention), and Appendix III (a computer program listing appendix containing the file and directory names for the files and directories on the compact disk comprising Appendix II).

[0068] The foregoing discussion provides examples that are intended to operate by way of example, not by limitation. There may be additions or changes to the processes and programs described above that fall within the scope and spirit of the invention. For example, any processing system may be used to execute the program instructions including systems having distributed processing networks, distributed data storage, and multiple I/O devices, in addition to the system shown in Fig. 2. Accordingly, the inventors hereby state their intention to rely upon the Doctrine of Equivalents to protect their full rights in the invention.